An Anchor Patch Based Optimization Framework For Reducing Optical Flow Drift in Long Image Sequences

Wenbin Li, Darren Cosker and Matthew Brown

Department of Computer Science, University of Bath, Bath, BA2 7AY UK

Abstract. Tracking through long image sequences is a fundamental research issue in computer vision. This task relies on estimating correspondences between image pairs over time where error accumulation in tracking can result in *drift*. In this paper, we propose an optimization framework that utilises a novel Anchor Patch algorithm which significantly reduces overall tracking errors given long sequences containing highly deformable objects. The framework may be applied to any tracking algorithm that calculates dense correspondences between images, e.g. optical flow. We demonstrate the success of our approach by showing significant tracking error reduction using 6 existing optical flow algorithms applied to a range of benchmark ground truth sequences. We also provide quantitative analysis of our approach given synthetic occlusions and image noise.

1 Introduction

Tracking a set of landmark points through multiple images is a fundamental research issue in computer vision. We define tracking here as the estimation of corresponding sets of vertices, pixels or landmark points between a reference frame and any other frame in the same image sequence. In the last decade, optical flow has become a popular approach for tracking through image sequences [1–3]. Compared with feature matching methods e.g. [4], optical flow provides sub-pixel accuracy and dense correspondence between a pair of images. In this paper, we focus in particular on improving tracking in image sequences using optical flow, and our contribution applies to this class of algorithm.

One of the main drawbacks of optical flow is *drift* [5]. Errors accumulated between frames over time results in movement away from the correct tracking trajectory. Between single image pairs, this problem may not be noticeable. However, accumulation when tracking across long sequences can be particularly problematic. Several authors have previously attempted to reduce optical flow *drift* in tracking. DeCarlo *et al.* [1] introduce contour information on a human face to improve tracking stability, while Borshukov *et al.* [2] employ manual correction. More recently, Bradley *et al.* [6] proposed an optimization method constrained by additional tracking information from multiview video sequences. Beeler *et al.* [7] then introduced the concept of anchor frames for human face

tracking. In this approach, the sequence is decomposed into several clips based on anchor images which are visually similar to a reference frame. Their optimization method shortens the tracking distance from reference frames to the target frame to help alleviate errors. However, their approach is domain specific (faces), and assumes that the entire face will return to a neutral expression (the anchor) several times throughout the sequence. In general, it is difficult to label anchor frames on general object sequences with large displacement motion e.g. waving cloth, as there is usually significant deformation between the reference frame and the other frames. In addition, repeated patterns are typically not global as observed in a face (return to a neutral expression). Rather, they occur in smaller local regions at intermittent intervals.

In this paper, we focus on tracking long video sequences using optical flow algorithms, and specifically concentrate on reducing *drift*. The general strategy of our approach is to shorten tracking distances for local regions throughout a long sequence. Our proposed framework combines long term feature matching with dense correspondence estimation. It may be applied to the tracking of general objects with large displacement motion, and results in a significant reduction in *drift*. We first detect *Anchor Frames* for a sequence (Section 4). This provides an initial set of start points for tracking the sequence. Our main contribution is extending this approach by proposing the concept of *Anchor Patches* (Section 5). These are corresponding points and patches throughout the sequence which are propagated directly from the reference frame. Our framework substantially reduces overall drift on a tracked image sequence, and may be applied to any optical flow algorithm in a straightforward manner. In our evaluation, we apply the proposed optimization framework on 6 popular optical flow estimation algorithms to illustrate it's applicability. We provide analysis of our method using 6 synthetic benchmark sequences (Section 7) generated using a method similar to [8], three of which are degraded by adding occlusion, gaussian noise and salt&pepper noise. In addition, we show its applicability on a popular publicly available real world facial sequence with manually annotated ground truth. We show that our proposed optimization framework significantly improves tracking accuracy and reduces overall drift when compared against the baseline optical flow approaches alone.

Our paper is organized as follows: In Section 2, an overview of our proposed optimization framework is outlined. Sections 3, 4, 5 and 6 give details of the four major steps in our framework. In Section 7, we evaluate our approach using 6 optical flow algorithms tested on 6 synthetic benchmark sequences and a real world facial sequence.

2 System Overview

Our proposed optimization framework reduces overall optical flow drift given long image sequences, and provides additional robustness against other issues such as large displacements and occlusions. The major procedure is shown in Table 1.The aim of our *Anchor Patch Optimization Framework* (APO) is accu-

Input: A reference frame, a triangle mesh and an image sequence
Step 1 (Sec. 3): Compute optical flow fields in both forward and backward directions **Step 2** (Sec. 4): Detect anchor frames and propagate the entire mesh to these frames **Step 3** (Sec. 5): Label anchor patches on non-anchor frames **Step 4** (Sec. 6): Track remaining patches from anchor frames to non-anchor frames
Output: A mesh tracked throughout the entire image sequence

Table 1. The major steps of the *Anchor Patch* optimization framework.

rately tracking a mesh denoted by $M_R = (V_R, E_R, F_R)$ from a reference frame I_R to every other frame I_i in the sequence. $M_i = (V_i, E_i, F_i)$ denotes the corresponding mesh on frame I_i. In the following sections, the four major steps are discussed in detail.

3 Step One: Computing The Optical Flow Field

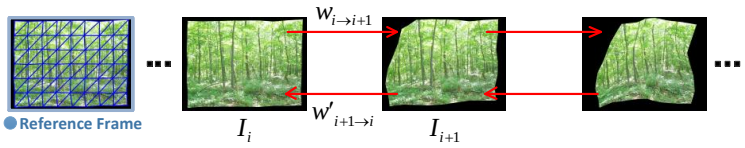


Fig. 1. Step One. The optical flow fields are computed in both forward ($\mathbf{w}'_{i\to i+1}$) and backward ($\mathbf{w}'_{i+1\to i}$) directions between every adjacent images pair in the sequence where the first frame is labelled as a reference frame.

The first step is to compute an optical flow field between every frame and its successor over a long video sequence in both forward and backward directions (Figure 1). In our evaluation, we consider application of our APO framework on a number of dense correspondence optical flow or tracking approaches, e.g. Brox *et al.* [5], Classic+NL [9] and ITV-L1 [10]. Let $\mathbf{w}_{i\to i+1}$ denote the optical flow field from frame I_i to frame I_{i+1}. Similarly we have $\mathbf{w}'_{i+1\to i}$ denoting the optical flow field from frame I_{i+1} to frame I_i in the backward direction. The optical flow field between frame I_i and I_j where $i < j$ (Forward direction), is denoted by $\mathbf{w}_{i\to j}$ as $\mathbf{w}_{i\to j} = \sum_{i<j} \mathbf{w}_{i\to i+1}$. Similarly, The optical flow field between frame I_j and I_i where $i < j$ (Backward direction), is denoted by $\mathbf{w}'_{j\to i}$ as $\mathbf{w}'_{j\to i} = \sum_{j>i} \mathbf{w}'_{j\to j-1}$.

In order to evaluate the optical flow at a specific pixel $\mathbf{X} = (x, y)^T$, an *Error Score* $E(w)$ is introduced, where $w = (u, v)^T$ is the optical flow vector at pixel $\mathbf{X}$. The pixel $\mathbf{X}$ in frame I_i is matched to pixel $\mathbf{X}' = (x', y')^T$ in frame I_{i+1} where $\mathbf{X}' = \mathbf{X} + w$. The *Error Score* $E(w)$ is calculated as the weighted *Root Mean Square* (RMS) error at a 3×3 pixel area centred on pixel $\mathbf{X}$ and $\mathbf{X}'$.

$$
\begin{aligned}
E(w) &= \sqrt{\frac{\alpha_1 d(x,y) + \alpha_2 d_{cross}(x,y) + \alpha_3 d_{diag}(x,y)}{\alpha_1 + \alpha_2 + \alpha_3}} \\
d_{diag}(x,y) &= d(x-1,y-1) + d(x+1,y+1) \\
&\quad + d(x-1,y+1) + d(x+1,y-1) \\
d_{cross}(x,y) &= d(x-1,y) + d(x+1,y) + d(x,y-1) + d(x,y+1) \\
d(x,y) &= |I_i(x,y) - I_{i+1}(x+u,y+v)|^2
\end{aligned} \tag{1}
$$

Where α_1, α_2 and α_3 are weights for controlling the contribution of each pixel in the 3×3 area. In our experiments, all these weights are set as $\alpha_1 = 1$, $\alpha_2 = 0.25$ and $\alpha_3 = 0.125$ which refer to the distance from the centre pixel $\mathbf{X}$ of the area. This *Error Score* is intended to evaluate the optical flow at a specific pixel. We also use it to evaluate feature matching scores later in our framework.

4 Step Two: Detecting Anchor Frames

Fig. 2. Step Two. The frames are detected as anchor frames (Red) because of the similar appearance to the reference (Blue). These anchor frames partition the entire sequence into several independent clips which allows tracking performing in parallel.

After obtaining our optical flow fields, anchor frames are then detected in a similar manner to Beeler *et al.* [7], with the difference that we employ *SIFT* for feature matching as opposed to *Normalised Cross Correlation* (NCC), and additionally use our *Error Score* function (Section 3) to evaluate matches. The main procedure is as follows (Figure 2):

- **Feature Capture.** A set of *SIFT* features S_R is detected in the reference frame I_R. Note that other features could be employed, but we select *SIFT* due to the general high accuracy and robustness.
- **Outlier Rejection.** The aim of this selection process is removing outliers from our feature matching. Correspondence matches of the *SIFT* feature set S_R between the reference frame I_R and the target frame I_i are performed. We select the matches which meet $|\mathbf{X} - \mathbf{X}'| < \tau$ where $\mathbf{X}$ is feature position in I_R, $\{\mathbf{X} \in S_R, \mathbf{X} = (x,y)^T\}$; $\mathbf{X}'$ is the corresponding feature position in I_i; τ is a threshold which is set as 30 pixels in our experiments. We find this

simple outlier rejection strategy sufficient for most of cases in our experiments (Section 7). More sophisticated outlier rejection method such as [11] could also be employed.

– **General Error Score.** The general error score is computed for every image as the average of the overall *Error Score* $E(w)$ (Equation (1)). Frames that contain the lowest general error score (below a specific threshold) are selected as anchor frames denoted I_A and the other frames are non-anchor frames. Figure 3 shows this process on our *Carton* benchmark sequence.

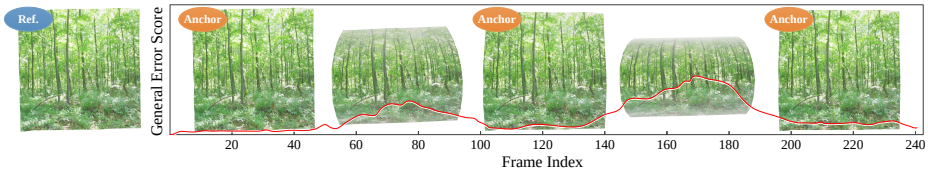


Fig. 3. The anchor frames are selected based on our general error score which is computed by comparing the reference frame to every other frame in our *Carton* benchmark sequence.

After detecting anchor frames which are visually similar to reference frame, these are used as a basis to partition the entire image sequence into several independent *clips*. This also allows computation in the next steps to be performed in parallel. In addition, the mesh M_R is propagated from the reference frame I_R to each anchor frame I_A using *SIFT* matches and a direct optical flow field between them. More detail can be found in section 6.1. The propagated mesh in an anchor frame is denoted $M_A = (V_A, E_A, F_A)$. Because of large displacement motion between anchor frames, and the fact that many images in a deformable sequence may not return to a reference point, these alone are typically insufficient to provide reliable tracking. In the next section, the *Anchor Patch* concept will be introduced to overcome this issue.

5 Step Three: Labeling Anchor Patches

The motivation of the original *Anchor Frame* method [7] is to provide multiple *Starting Points* for tracking. Since error accumulates, the technique is intended to reduce overall error accumulation across long image sequences. However, as mentioned in the previous section, large displacement motion contains high *degrees of freedom* (DoF), meaning that most images in a video sequence will have significant visual differences from the reference frame.

The central observation in long image tracking is that local spatial patterns throughout a sequence will be repeated - i.e. part of a cloth might return to the same position several times throughout a video. We take advantage of these

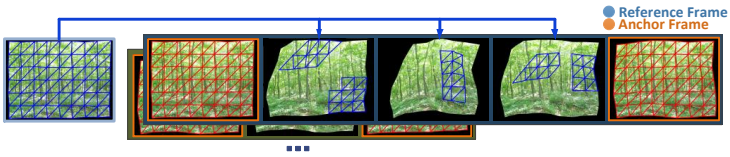


Fig. 4. Step Three. Anchor patches (blue patches) are label on non-anchor frames within every clip using *SIFT* feature matching and *Barycentric Coordinate Mapping* between reference frame and non-anchor frame.

repeating regions in order to track between shorter segments, and thus alleviate error accumulation. As opposed to an entire image from a sequence acting as an anchor, an *Anchor Patch* is defined as a set of individual vertices or an area of pixels in the non-reference frame (any other frame in the sequence), which are highly correlated to a specific part of the reference. The benefit of using anchor patches is to provide additional information for correcting errors when tracking using optical flow. This technique can also reduce the impact of a low-quality anchor frame (i.e. one which is too dissimilar from the reference frame). Before anchoring patches on non-anchor frames, we first obtain a set of high-quality *SIFT* feature matches between the reference frame and non-anchor frames, i.e. those which are not already labelled as the reference frame, or an existing anchor frame. This process proceeds as follows:

- **Feature Capture.** Similar to *Step Two* (Section 4), *SIFT* is employed to detect a feature set S_R in the reference frame I_R.
- **Matching Selection.** We use the *VLfeat* matching approach [12] to perform correspondence matching of *SIFT* feature sets S_R to feature set S_i of the non-anchor frames I_i. Matches are selected where the *Error Score* (Equation (1)) is below a predefined threshold. This process generates a matches set $\mathbf{m}_{R\to i}$ between the reference frame I_R and non-anchor frame I_i.

The set of matches $\mathbf{m}_{R\to i}$ is used as our initial basis for anchoring patches on non-anchor frames. In order to obtain final anchor patches, *Barycentric Coordinate Mapping* and *Error Refinement* are applied as follows:

Barycentric Coordinate Mapping We wish to determine the pixel position in a non-anchor frame which corresponds to the position of a vertex on the reference mesh M_R in I_R. These correspondences provide our baseline for stable tracking throughout the image sequence. Figure 5 illustrates the process of anchoring patches where $v = (x, y)^T$ denotes a vertex in M_R; $f_* = (x_*, y_*)^T$, and denotes *SIFT* features in the reference frame I_R. Similarly, $f'_* = (x'_*, y'_*)^T$ denotes *SIFT* features in a non-anchor frame I_i. For the non-anchor frame I_i, we have $\{f_k \to f'_k \in \mathbf{m}_{R\to i}, k = 1, 2, 3 \dots\}$ which denotes previously obtained corresponding *SIFT* feature matches. We wish to calculate the new vertex position $v' = (x', y')^T$ in the non-anchor frame I_i. We do this by searching for the three

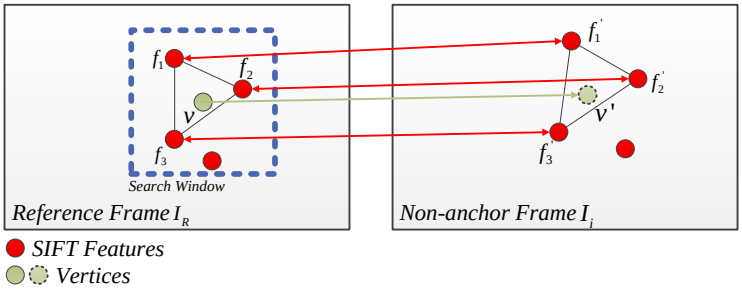


Fig. 5. Anchoring patches using *Barycentric Coordinate Mapping* and *SIFT* features.

nearest *SIFT* features f_* in a small 5×5 search window centred on the vertex of interest v. Next, v' is calculated by solving the *Barycentric Coordinate Mapping* equations as:

$$\begin{bmatrix} f_1 & f_2 & f_3 \\ f'_1 & f'_2 & f'_3 \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix} = \begin{bmatrix} v \\ v' \end{bmatrix} \tag{2}$$

Where β_* are intermediate variables that satisfy $\beta_1 + \beta_2 + \beta_3 = 1$. In practice we found this technique to provide an accurate transformation when applied to small region (5×5 pixel block). However, more sophisticated (although slower) interpolation methods such as *Thin-plate Spline* could also be used. The process is performed on every vertex in M_R.

Error Refinement After *Barycentric Coordinate Mapping*, candidate anchor patches denoted by v'_* are obtained in non-anchor frames I_i. We also have matches $v_* \rightarrow v'_*$, the strength of which can be evaluated using our error equation (1). Using this error, we select final anchor patches in a non-anchor frame I_i using $\{P(v'_*)|E(v_* \rightarrow v'_*) < \eta\}$ where η is a predefined threshold.

6 Step Four: Mesh Propagation

The objective of our optimization framework is to track a mesh M_R from the reference frame to every other frame in an image sequence. Given tracking information from the following sections, this process is separated into two steps: first, the mesh M_R is propagated from reference frame to anchor frames (section 4 and 6.1). Second, the propagated mesh M_A is propagated from anchor frames to the non-anchor frames within the clip (section 6.2).

6.1 Propagating from the reference frame to anchor frames

The mesh propagation process from the reference frame to the anchor frame is as follows:

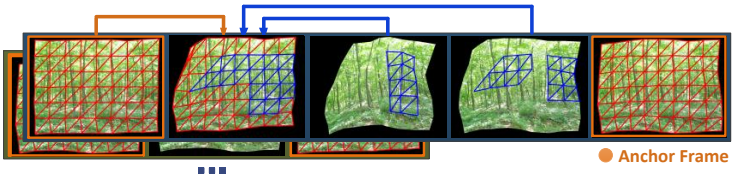


Fig. 6. Step Four. Tracking other patches from the anchor frame and nearest anchor patches within a clip where the blue patches are anchor patches, selected from *Nearest Anchor Patch*.

- **Computing the optical flow field.** The optical flow field $\mathbf{w}_{R\to A}$ directly between the reference frame to the anchor frame is computed.
- **Matching selection.** For every vertex in M_R, *high error matches* are eliminated (see Error Refinement).
- **Barycentric Coordinate Mapping.** *Barycentric Coordinate Mapping* is applied to low error matches.

After this stage, information for every vertex in M_R is established from the reference frame to the anchor frame.

6.2 Propagating from anchor frames to non-anchor frames

The entire image sequence is partitioned into clips which are bound by different anchor frames. The propagation process can be individually performed within these clips in parallel. As the non-anchor frames contain anchor patches, this improves overall tracking stability within these clips. Figure 6 illustrates this concept. In order to use anchor patches in this process, the concept of a *Nearest Anchor Patch* is also defined. For vertex v in M_A, the *Nearest Anchor Patch* of v on frame I_i is the anchor patch $\left\{v'_{i+k}|v \to v'_{i+k}\right\}$ on non-anchor frame I_{i+k} which is nearest to I_i in the image sequence. Figure 7 shows an example where frame I_{i+k} is the frame which is nearest to frame I_i in image sequence and contains anchor patch v'_{i+k} matching to v in anchor frame I_A. The main tracking procedure proceeds as follows:

- **Mesh propagation.** In order to establish tracking information between anchor frames and non-anchor frame, the mesh M_A is first propagated from anchor frame I_A to non-anchor frames I_i using the previously calculated optical flow field $\mathbf{w}_{A\to i}$ from *Step One* (Section 3).
- **Anchor patches propagation.** The *Nearest Anchor Patch* of each vertex v in M_A is searched through the whole clip then propagated to non-anchor frame I_i using the optical flow field in the forward $\mathbf{w}_{*\to i}$ or backward $\mathbf{w}'_{i+k\to i}$ direction.
- **Conflict eliminating.** After propagating the mesh and nearest anchor patches to non-anchor frame I_i, there may be position conflict on some of the

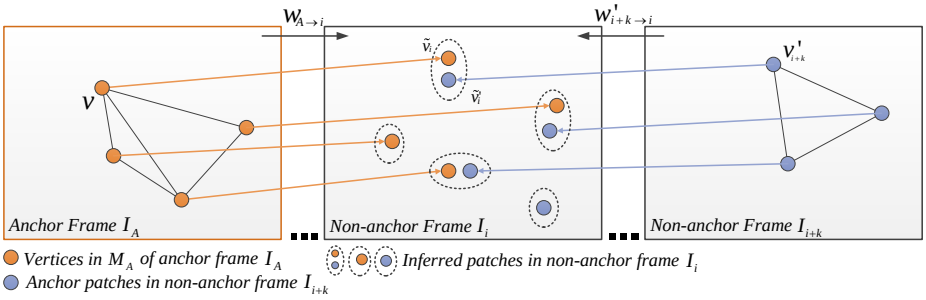


Fig. 7. Vertex conflict can happen when mesh and anchor patches are propagated to target frame I_i. Here v'_{i+k} is an anchor patch that is strongly matched to v.

propagated vertices. As shown in Figure 7, $\tilde{v}_i$ and $\tilde{v}'_i$ are not in the same desired position. In order to eliminate the conflict, the position of $\{v_i | v \rightarrow v_i\}$ matching to v can be calculated using the sum of all weighted candidate positions e.g. $\tilde{v}_i$ and $\tilde{v}'_i$ (Eq.3) based on the *Error Score*.

$$v_i = \frac{E(v \rightarrow \tilde{v}'_i)\tilde{v}_i + E(v \rightarrow \tilde{v}_i)\tilde{v}'_i}{E(v \rightarrow \tilde{v}'_i) + E(v \rightarrow \tilde{v}_i)} \tag{3}$$

Due to the fact that the anchor frames divide the overall sequence into smaller clips, this allows the mesh propagation in between to be calculated in parallel. In the next section we perform an evaluation of our framework.

7 Evaluation

We evaluate APO with a range of 6 popular optical flow estimation methods which are publicly available from the *Middlebury Evaluation System* [13]. *Combined local-global Optical Flow* (CLG-TV) [14], *Large Displacement Optical Flow* (LDOF) [5] and Classic+NL [9] are state of the art while the *Horn and Schunck* (HS) [15], *Black and Anandan* (BA) [16, 9], *Improved TV-L1* (ITV-L1) [10] are classic optical flow frameworks and also widely used. CLG-TV is a high speed approach that uses a combination of bilateral filtering and anisotropic regularization and also one of the top three algorithms in the normalized interpolation error test from Middlebury. LDOF is an integration of rich feature descriptors and variational optical flow and one of best current optical flow estimation algorithms for large displacement motion. Classic+NL provides high performance in the Middleburry evaluation by formalizing the median filtering heuristic and Lorentzian penalty as explicit objective functions in an *improved TV-L1* framework. The HS method is a pioneering technique optical flow. BA provides improvements to the HS framework by introducing robust quadratic error formulation. ITV-L1 is a recent and increasingly popular optical flow framework which uses a similar numerical optimization scheme to Classic+NL. Our choice of a

	Information of the Benchmark Sequences						
	Original	Occlusion	Guass.N	S&P.N	Carton	Serviette	Frank
Image Size (pix.)	500 × 500	500 × 500	500 × 500	500 × 500	1024 × 768	1024 × 768	720 × 576
Sequence Length	237	237	237	237	266	307	300
Annotation Points	160	160	160	160	81	63	68
Avg. Feature Amount	364.80	358.32	566.13	1276.50	2498.01	3315.49	2071.11

Table 2. An overview of the benchmark sequences in our evaluation. That includes 4 attributes of image size (pixel), sequence length, number of ground truth annotation points per frame and average SIFT feature amount per frame.

mixture of newer, state of the art methods, with older traditional approaches, is to highlight the fact that irrespective of the approach used, our APO framework provides significantly improved tracking in all cases.

For our evaluation, we compare the optical flow estimation methods previously mentioned – with and without our optimization framework – on 7 long benchmark sequences with ground truth. Table 2 gives an overview of the benchmark sequences used in our evaluation. In previous work Garg *et al.* released to the community a set of ground truth data for evaluating optical flow algorithms over long sequences. This is as opposed to the Middlebury dataset, which just considers optical flow between pairs of images, and is therefore not applicable to our framework. The sequences of Garg *et al.* contains 60 frames and are generated using interpolated dense *Motion Capture* (MOCAP) data from real deformations of a waving flag [17]. We use the same MOCAP data to generate a long video sequence and three other degraded sequences, each of which contains 237 frames of size 500×500 pixels. The three degraded sequences are generated in order to test the robustness of our APO framework under different image conditions. They are generated by individually adding synthetic occlusions, gaussian noise and salt & pepper noise with the same parameters described in [8]. In order to increase the diversity of the sequences, we include three other sequences. One is a *Talking Face Video* (Frank) sequence which contains 300 frames with 68 ground truth annotation points per frame. The other two are also synthetic benchmark sequences generated using MOCAP data of Salzmann *et al.* [18] from the carton and serviette deformations. One contains 266 frames of size 1024×768 while the other contains 307 frames of the same image size. In addition, we also consider the effect of the number of SIFT features detected in the frame, and how this affects overall tracking stability of the APO framework. All optical flow algorithms are applied with default parameter settings from their original papers.

Our baseline optical flow based tracking strategy – for each of the above algorithms – is performed as follows: First, the optical flow field is computed (in forward direction) for every pair of adjacent frames in the sequence. We then mark the initial tracking points in the first frame using the same ground truth points in the same frame of the sequence (Table 2). The correspondent points in the next frame are computed based on the optical flow field in between. This process is repeated until correspondent landmark points are obtained in every frame of the sequence. The *Root Mean Square (RMS) Endpoint Error (EE)* [13] is then calculated against the ground truth annotation points. We then apply

Methods	Average **RMS EE** (pix)						
	Original	Occlusion	Guass.N	S&P.N	Carton	Serviette	Frank
BA [16]	6.14	8.03	11.02	7.79	10.56	5.18	17.57
BA + ***APO***	1.72^{2}	1.91^{2}	$\mathbf{7.89}^{1}$	$\mathbf{5.04}^{1}$	2.77	$\mathbf{1.56}^{1}$	6.60
CLG-TV [14]	8.59	10.93	20.28	33.93	28.94	32.17	19.29
CLG-TV + ***APO***	2.25	2.97	12.31	18.99	6.95	9.43	7.05
HS [15]	29.16	30.44	29.74	29.43	27.69	37.90	31.27
HS + ***APO***	11.68	12.88	17.79	17.21	10.25	10.03	14.19
LDOF [5]	6.21	6.39	16.24	24.14	6.33	5.51	14.73
LDOF + ***APO***	1.75^{3}	$\mathbf{1.67}^{1}$	11.65	13.12	$\mathbf{1.18}^{1}$	1.84^{2}	$\mathbf{3.12}^{1}$
Classic+NL [9]	7.07	10.61	12.65	9.50	5.72	6.62	17.32
Classic+NL + ***APO***	2.15	3.18	8.31^{2}	6.46^{2}	1.34^{2}	2.03^{3}	3.44^{2}
ITV-L1 [10]	5.73	8.25	17.29	14.49	5.34	7.11	17.91
ITV-L1 + ***APO***	$\mathbf{1.50}^{1}$	2.33^{3}	9.53^{3}	7.70^{3}	1.70^{3}	2.36	3.69^{3}

Table 3. Average RMS *Endpoint Error* (EE) comparison of different methods with our optimization framework on the benchmark sequences.

our APO framework using the same optical flow fields.). Note that the parameter values relevant to the APO framework are initially and experimentally selected, but then remain constant in all our evaluations.

Table 3 shows the measurement of average *RMS EE* in pixels over all the frames of the sequences. We highlight the top three best *RMS EE* measures for each sequence using superscripts next to different values. Notice that APO significantly reduces the *RMS EE* compared to the baseline optical flow methods. Our optimization framework yields the best *RMS EE* measure in all the cases. For instance, *ITV-L1* with APO performs the best in sequence *Original* while *LDOF* with APO yields the best result in sequence *Frank*. We also observe that although in the *Guass.Noise* and *S&P.Noise* sequences the improvement is less than in the unaltered sequences, the overall result is still an improvement with the addition of APO.

While we concern ourselves primarily with tracking over long sequences in this paper, we also consider here shorter sequences. In Table 4, the average *RMS EE* measures of various methods are compared on the first 30 frames of our benchmark sequences. We observe similar *RMS EE* measures as in the long sequence case (Table 3). The APO framework significantly increases the tracking accuracy – outperforming the baseline tracking methods in all cases even given degradation (e.g. *Gauss.Noise* and *S&P.Noise*). Moreover, the *BA* with APO is also observed to overfit in the noisy sequences while *Classic+NL* with APO yields the best measures in both sequences of *Gauss.Noise* and *S&P.Noise*.

We also evaluate the effect on tracking accuracy by varying the number of selected features. Different numbers (50% and 0%) of features are randomly selected from the initial full detection feature set before performing *Anchor Patch* detection. Information on our total number of features can be found in Table 2, e.g. there are 364.80 features averagely on each frame of the sequence *Original*. Table 5 shows an average *RMS EE* comparison given various numbers of features. We observe that *RMS EE* improves given more features in all cases. Another interesting observation is that our optimization framework provides lower *RMS EE* against the baseline tracking strategy even given sparse or no features (0%

Methods	Average **RMS EE** (pix) on the First 30 Frames						
	Original	Occlusion	Guass.N	S&P.N	Carton	Serviette	Frank
BA [16]	1.57	1.72	3.87	2.71	2.37	1.56^{3}	8.76
BA + *APO*	1.41^{3}	1.65	3.66^{3}	2.13^{2}	2.17	$\mathbf{1.13}^{1}$	5.40
CLG-TV [14]	2.40	2.60	6.71	8.77	8.10	5.54	8.60
CLG-TV + *APO*	2.10	2.24	6.53	8.39	4.79	5.11	7.35
HS [15]	33.67	35.70	35.05	34.50	26.16	22.08	12.76
HS + *APO*	16.11	16.32	13.78	19.37	9.78	6.33	9.19
LDOF [5]	2.38	2.37	3.96	4.03	3.90	2.52	8.51
LDOF + *APO*	1.15^{2}	$\mathbf{0.97}^{1}$	3.75	2.66	$\mathbf{0.89}^{1}$	1.44^{2}	$\mathbf{2.82}^{1}$
Classic+NL [9]	1.63	1.76	3.61^{2}	2.51^{3}	2.18	1.75	8.77
Classic+NL + *APO*	1.51	1.33^{2}	$\mathbf{3.54}^{1}$	$\mathbf{1.99}^{1}$	1.24^{2}	1.68	3.70^{3}
ITV-L1 [10]	1.55	1.76	6.27	5.07	2.37	2.01	9.22
ITV-L1 + *APO*	$\mathbf{0.99}^{1}$	1.31^{2}	5.77	4.65	1.69^{3}	1.71	3.48^{2}

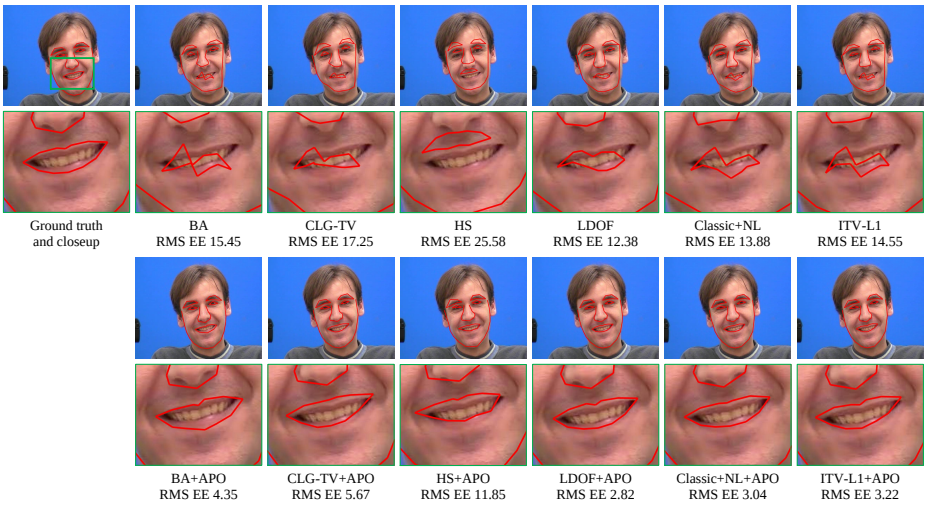
Table 4. Average RMS *Endpoint Error* (EE) comparison of different methods with our optimization framework on the first 30 frames of the benchmark sequences.

Methods	Average **RMS EE** (pix) on Different Feature Distributions						
	Original	Occlusion	Guass.N	S&P.N	Carton	Serviette	Frank
BA [16], **No *APO***	6.14	8.03	11.02	7.79	10.56	5.18	17.57
***APO*, 100% Feature**	1.72^{2}	1.91^{2}	$\mathbf{7.89}^{1}$	$\mathbf{5.04}^{1}$	2.77	$\mathbf{1.56}^{1}$	6.60
***APO*, 50% Feature**	3.64	4.71	8.06	6.12	5.89	2.98	10.63
***APO*, 0% Feature**	5.12	6.44	9.23	7.21	8.69	4.35	12.69
CLG-TV [14], **No *APO***	8.59	10.93	20.28	33.93	28.94	32.17	19.29
***APO*, 100% Feature**	2.25	2.97	12.31	18.99	6.95	9.43	7.05
***APO*, 50% Feature**	4.86	6.51	14.39	22.72	15.36	19.91	12.00
***APO*, 0% Feature**	6.94	9.11	16.83	26.03	23.57	24.03	15.07
HS [15], **No *APO***	29.16	30.44	29.74	29.43	27.69	37.90	31.27
***APO*, 100% Feature**	11.68	12.88	17.79	17.21	10.25	10.03	14.19
***APO*, 50% Feature**	18.13	20.28	20.66	19.91	17.39	25.99	23.45
***APO*, 0% Feature**	24.73	27.11	23.97	23.40	24.09	33.11	29.17
LDOF [5], **No *APO***	6.21	6.39	16.24	24.14	6.33	5.51	14.73
***APO*, 100% Feature**	1.75^{3}	$\mathbf{1.67}^{1}$	11.65	13.12	$\mathbf{1.18}^{1}$	1.84^{2}	$\mathbf{3.12}^{1}$
***APO*, 50% Feature**	3.21	3.09	12.18	15.02	2.90	3.74	8.66
***APO*, 0% Feature**	5.08	5.24	14.11	18.46	5.45	4.89	11.76
Classic+NL [9], **No *APO***	7.07	10.61	12.65	9.50	5.72	6.62	17.32
***APO*, 100% Feature**	2.15	3.18	8.31^{2}	6.46^{2}	1.34^{2}	2.03^{3}	3.44^{2}
***APO*, 50% Feature**	4.00	6.39	9.48	7.33	3.89	4.00	10.14
***APO*, 0% Feature**	5.96	7.78	11.64	8.98	4.78	6.00	13.27
ITV-L1 [10], **No *APO***	5.73	8.25	17.29	14.49	5.34	7.11	17.91
***APO*, 100% Feature**	$\mathbf{1.50}^{1}$	2.33^{3}	9.53^{3}	7.70^{3}	1.70^{3}	2.36	3.69^{3}
***APO*, 50% Feature**	3.59	5.17	10.93	8.47	3.41	5.00	10.11
***APO*, 0% Feature**	4.77	6.92	12.50	10.31	4.43	5.95	14.29

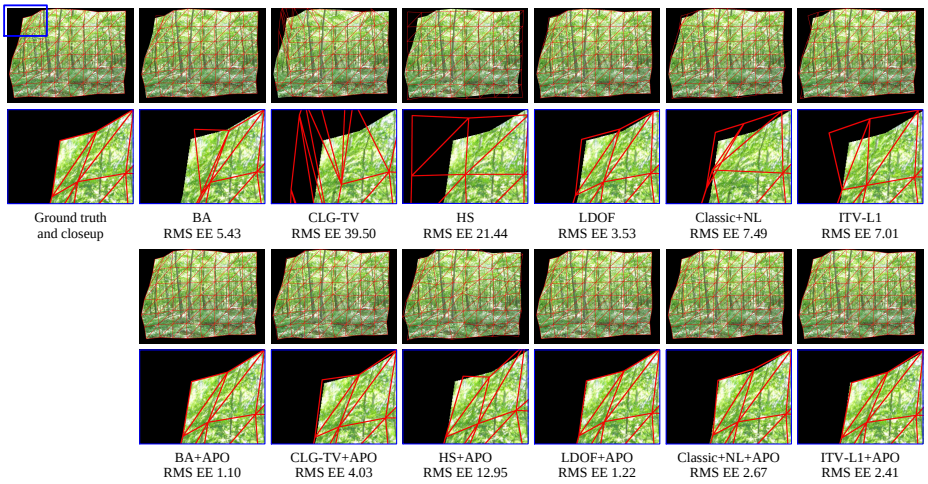
Table 5. Average RMS *Endpoint Error* (EE) comparison on the benchmark sequences with varying feature distributions.

feature). Note that in this case, our APO framework defaults to using an optical flow method with just the *Anchor Frame* approach [7]. Also note – for example by comparing to Table 3 – that this indicates that the APO framework also provides significant tracking improvement over using anchor frames alone.

We also make the visual comparisons on two of our sequences, *Frank* and *Serviette*. The former is real world sequence with ground truth annotation points, while the latter is synthetic sequence overlaid with a ground truth mesh. In Figure 8, we observe noticeable *drift* problems given the baseline optical flow tracking strategy. Also note that the APO framework significantly reduces the *drift* problem.



(a) Visual comparison of different methods on the frame 88 of the sequence *Frank*.



(b) Visual comparison of different methods on the frame 192 of the sequence *Serviette*.

Fig. 8. Visual comparison and *RMS EE* measures on sequences of *Frank* and *Serviette*.

8 Conclusion

In this paper, we have presented an optimization framework based on *Anchor Patches* for improving mesh or sparse point set tracking during long video image sequences. Our optimization framework anchors image regions throughout the sequence to mitigate the effect of *Error Accumulation* and *Drift*. In our evaluation, we have compared APO combined with 6 popular optical flow estimation algorithms against baseline tracking on 7 benchmark sequences. This includes 6 synthetic benchmark sequences with real world deformation and 1 real world

sequence. We have demonstrated that APO provides significant tracking improvements for dense correspondence based tracking on long video sequences than using baseline optical flow tracking alone.

References

1. DeCarlo, D., Metaxas, D.: The integration of optical flow and deformable models with applications to human face shape and motion estimation. In: Computer Vision and Pattern Recognition (CVPR), IEEE (1996) 231–238
2. Borshukov, G., Piponi, D., Larsen, O., Lewis, J., Tempelaar-Lietz, C.: Universal capture: image-based facial animation for the matrix reloaded. In: ACM SIGGRAPH 2005 Courses, ACM (2005) 16
3. Ezzat, T., Geiger, G., Poggio, T.: Trainable videorealistic speech animation. ACM Transactions on Graphics (TOG) **21** (2002) 388–398
4. Lowe, D.: Distinctive image features from scale-invariant keypoints. International journal of computer vision **60** (2004) 91–110
5. Brox, T., Malik, J.: Large displacement optical flow: Descriptor matching in variational motion estimation. IEEE Transactions on Pattern Analysis and Machine Intelligence **33** (2011) 500–513
6. Bradley, D., Heidrich, W., Popa, T., Sheffer, A.: High resolution passive facial performance capture. ACM Transactions on Graphics (TOG) **29** (2010) 41
7. Beeler, T., Hahn, F., Bradley, D., Bickel, B., Beardsley, P., Gotsman, C., Sumner, R., Gross, M.: High-quality passive facial performance capture using anchor frames. In: ACM Transactions on Graphics (TOG). Volume 30., ACM (2011) 75
8. Garg, R., Roussos, A., Agapito, L.: Robust trajectory-space tv-l1 optical flow for non-rigid sequences. Energy Minimazation Methods in Computer Vision and Pattern Recognition (2011) 300–314
9. Sun, D., Roth, S., Black, M.: Secrets of optical flow estimation and their principles. In: Computer Vision and Pattern Recognition (CVPR). (2010) 2432 –2439
10. Wedel, A., Pock, T., Zach, C., Bischof, H., Cremers, D.: An improved algorithm for tv-l1 optical flow. Statistical and Geometrical Approaches to Visual Motion Analysis (2009) 23–45
11. Pizarro, D., Bartoli, A.: Feature-based deformable surface detection with self-occlusion reasoning. International Journal of Computer Vision (2012) 1–17
12. Vedaldi, A., Fulkerson, B.: VLFeat: An open and portable libraryof computer vision algorithms (2008)
13. Baker, S., Scharstein, D., Lewis, J., Roth, S., Black, M., Szeliski, R.: A database and evaluation methodology for optical flow. International Journal of Computer Vision **92** (2011) 1–31
14. Drulea, M., Nedevschi, S.: Total variation regularization of local-global optical flow. In: Intelligent Transportation Systems (ITSC), IEEE (2011) 318–323
15. Horn, B., Schunck, B.: Determining optical flow. Artificial intelligence **17** (1981) 185–203
16. Black, M., Anandan, P.: The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. Computer vision and image understanding **63** (1996) 75–104
17. White, R., Crane, K., Forsyth, D.: Capturing and animating occluded cloth. In: ACM Transactions on Graphics (TOG). Volume 26., ACM (2007) 34
18. Salzmann, M., Hartley, R., Fua, P.: Convex optimization for deformable surface 3-d tracking. Computer Vision, IEEE International Conference on **0** (2007) 1–8